PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants:	MOLDOVYAN et al.	Attorney Docket:	P65855US0
Serial No.:	09/622,047	Group Art Unit:	2132
Filing Date:	August 23, 2000	Examiner:	Benjamin E. LAMIER

For: METHOD FOR THE BLOCK-ENCRYPTION OF DISCRETE DATA

TRANSMITTAL

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith is the APPEAL BRIEF in the above captioned application in support the Notice of Appeal to the Board of Patent Appeals and Interferences filed on March 13, 2007.

The fee has been calculated as shown below:

Claims Remaining After Amendment		Highest Number Previously Paid For		Present Extra	Small Entity Rate	Small Entity Addit. Fee		Other Than A Small Entity Rate	Other Than A Small Entity Addit. Fee
Total	3	-	20	= 0	x25	= $	(or)	x 50	= $
Indep.	1	-	3	= 0	x100	= $		x 200	= $
Appeal Brief Fee						$			$500.00
Total Additional Fee						$			$500.00

XX Credit Card Payment in the amount of $500.00 is attached.

XX If a Petition for Extension of Time is necessary and the Petition and/or the check is not enclosed, this will act as the Petition and applicant herewith petitions the Commissioner to extend the time for response and charge any fees necessary under 37 CFR 1.17 (a)(1)-(5) to Deposit Account No. 06-1358. The Commissioner is also authorized to charge payment of any other additional fees associated with this communication or credit any overpayment to Deposit Account No. 06-1358. A duplicate copy of this sheet is attached.

JACOBSON HOLMAN, PLLC

Dated: May 8, 2007
400 Seventh Street, N. W.
Washington, D.C. 20004-2201
JCH/JC

By: (Reg. #58,140) for
John C. Holman
Reg. No. 22,769

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	Confirmation Number: 4150
MOLDOVYAN et al.	Attorney Docket: P65855US0
Serial No. 09/622,047	Group Art Unit: 2132
Filed: August 23, 2000	Examiner: Benjamin E. LANIER

For: METHOD FOR THE BLOCK-ENCRYPTION OF DISCRETE DATA

APPEAL BRIEF UNDER 37 C.F.R. 41.37

Mail Stop Appeal Brief -Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an appeal from the Examiner's Final Rejection dated September 14, 2006 of Claims 1, 3, and 5. A Notice of Appeal was filed on March 13, 2007, together with a Request for three-month extension of time.

I. REAL PARTY IN INTEREST

The real part in interest is OTKRYTOYE AKTSIONERNOYE OBSCHESTVO by virtue of the assignment recorded at Reel/Frame 011071/0187-0188 in this application on August 23, 2000.

II. RELATED APPEALS AND INTERFERENCES

Appellants, Appellants' representative and their assignee are not aware of any other appeals or interferences which will directly affect or be directly affected by or having a bearing on the decision of the Board of Patent Appeals and Interferences ("Board") in this appeal.

III. STATUS OF THE CLAIMS

The appealed claims are Claims 1, 3, and 5, which are currently pending in this application. Claims 1, 3, and 5 stand rejected under 35 U.S.C. § 102 (b) as allegedly being anticipated by Schneier (Bruce Schneier, Applied Cryptography, 1996, John Wiley & Sons, pages 270 – 273, copies are enclosed in the Evidence Appendix). A copy of the claims on appeal appears in the attached Claims Appendix.

IV. STATUS OF AMENDMENT

The pending claims were not amended in the Response to the Final Rejection filed on January 5, 2007.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The present invention is related to cryptographic methods for encrypting data that can be used in the fields of electronic communication and computer technology. According to the known methods, the block data encryption is achieved by generating an encryption key in the form of a plurality of subkeys, splitting data block into subblocks, and alternately convert the data subblock by using a cyclic offsetting operation, modulo 2 addition operation, and modulo 2^{32} addition operation. The subkeys are used according to a fixed schedule. The subkey is independent of the data block. In the U.S. standard DES (National Bureau of Sdtandards, Data Encryption Standard Federal Information Processings Standard Publication 46, January 1997), the data sublocks are converted according to a secret key control. In this method, at some fixed encryption round number, a fixed subkey is used for all data subblocks. These methods have many drawbacks, such as low encryption rate and insufficient resistance to differential and linear cryptanalysis (see pages 1 – 2). The present invention provide a method for block encryption of discrete data, comprising the steps of:

generating an encryption key in the form of a set of subkeys, breaking down a data block into N ≥ 2 data subblocks and alternately converting said data subblocks by performing a two-place operation on the data subblock and the subkey, wherein, prior to carrying out said two-place operation on an i-th data subblock and a subkey, an operation of permuting subkey bits is performed on the subkey depending on the value of a j-th data subblock, where i ≠ j (see pages 3, Examples 1 – 3 on pages 5 – 11, Figs. 1 – 6, Claim 1). In addition, according to the claimed method, an operation of cyclic offsetting subkey bits depending on the value of the j-th data subblock is used as the j-th data subblock-dependent operation of permuting subkey bits (see Claim 3, Examples 1 – 3 on pages 5 – 11, Figs. 1 – 6). Furthermore, according to the claimed method, the operation of permuting subkey bits is performed on one of said set of subkeys depending on the value of the j-th data subblock, where i ≠ j, and the value of another subkey (see Claim 5, Examples 1 – 3 on pages 5 – 11, Figs. 1 – 6). The claimed methods overcome the drawbacks of the known methods (see page 12).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether Claims 1, 3, and 5 are properly rejected under 35 U.S.C. § 102 (b) as being anticipated by Schneier (Bruce Schneier, Applied Cryptography, 1996, John Wiley & Sons, pages 270 – 273).

VII. ARGUEMENT

Claims 1, 3, and 5 Are Not Anticipated by Schneier Reference Because It Does Not Teach or Suggest All the Limitations of These Claims

The final rejections of the application are based on 35 U.S.C. 102 (b). According to the U.S. patent law and according to MPEP 2131, "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

In the Final Office Action, the Examiner indicated that Schneier discloses key bit permutation operation depending on the data being converted. The issue is how to correctly interpret the disclosure of Schneier by a person of ordinary skill in the art.

Applicant respectfully submits that, Schneier, when describing algorithm DES (US Data Encryption Standard), does not disclose any feature of converting a subkey depending on data being converted. Fig.12.1 of Schneier shows a general diagram of data conversion in accordance with the encryption algorithm DES, which includes 16 rounds of conversion. In each round of conversion, based on a right subblock R and subkey K, function $f = f(R, K)$ is calculated, after which a left subblock L is converted by performing on it the operation XOR: $L: = L \oplus f(R, K)$, where ": =" is the designation of assignment operation. Between the preceding and subsequent encryption rounds, the subblocks are transposed (swapped). Thus, it is important to ascertain, how conversion $f(R, K)$ is performed. In the Examiner's view, in calculating $f(R, K)$, the operation of permuting subkey bits **depending on data being converted** is used. However, Applicant's detailed study of Schneier has shown that this is not the case. More specifically, Schneier shows that the procedures of calculating the function $f(R, K)$ includes consecutive performing the following operations:

- operation of broadening L 32-bit data subblock to X broadened 48-bit data subblock;
- conversion of the broadened subblock by means of its addition with 48-bit subkey $X : = X \oplus K$ (before this step, no conversions depending on the data block being converted have been performed on this round subkey, i.e. **no permuting subkey bits depending on data has been performed)**;
- performing a cascade of substitution operations of 6x4 size implementing the substitution operation S_{6x4}, as a result of which the broadened 48-bit subblock is converted into 32-bit binary vector $Z : Z = S_{6x4}(X);$
- performing the transmutation operation $\boldsymbol{P}$ which consists in a fixed permutation of the vector $\boldsymbol{Z}$ bits, i.e. permutation of the vector $\boldsymbol{Z}$ bits is performed independently of the value of some data subblock but always in the same

manner, as prescribed by the Schneier reference. After performing operation ***P***, the value of *f* (*R*, *K*) is obtained, i.e. we have *f* (*R*, *K*) = ***P(Z)***.

The above detailed operations have revealed that, in forming the value of *f* (*R*, *K*) in the DES algorithm, **the operation of permuting subkey bits depending on data being converted is not used**. The vector ***Z*** bit transmuting operation performed in the cited method of block encryption (algorithm DES) is fixed and is performed **regardless of data being converted**. Schneier confirms this by describing the operation ***P*** in the section "The P-box permutation".

The Applicant has also studied the procedures of working-out round subkeys K_1, K_2, ..., K_{16}. According to Schneier, round keys K_1, K_2, ..., K_{16} are generated by means of converting a secret key, on which an operation of fixed transmuting key bits is performed, which depends on the round number but **does not depend on data subblock being converted**. For a given round, this operation of transmuting key bits is the same for all different datablocks being converted. Following the above fixed key bit transmutation, a fixed compressing key bits transmutation is performed, a result of which is this value of the current round subkey. Fixed compressing key bits transmutation **does not depend on the data subblock being converted** and remain always the same. Thus, the procedures of forming round keys of DES algorithm also lacks the feature of "prior to carrying out said two-place operation on an i-th subblock and a subkey, an operation of permuting subkey bits is performed on the subkey **depending on the value of a j-th data subblock**, where i ≠ j."

Thus, the Examiner has incorrectly interpreted the conversion operation *f* appeared on page 270 and in Fig. 12.1 of Schneier as an operation of permuting bits of subkey K_1. In fact, the conversion operation *f* is not a permutation operation. This is further supported by the following evidence.

The number of "one" bits and the number of "zero" bits in a binary vector produced as a result of combined conversion of subkey K_1 and data subblock R_0 (page 270 and Fig.

12.1) is typically **not equal to** the number of "one" and "zero" bits in either subkey K_1 or in data subblock R_0. However, as a result of performing the operation of bit permutation, the number of "one" bits and the number of "zero" bits in the output binary vector **is always equal to** the number of "one" bits and the number of "zero" bits, respectively, in the input binary vector. This is because, in performing the operation of permutation, the bits are permuted but not inverted. The fact that the conversion operation *f* is not an operation of permutation, is also supported by B. Schneier, A. Menezes (see page 225 of Menezes A.J., Vanstone S.A., Handbook of Applied Cryptography, CRC Press, 1996, previously submitted in the response filed on August 22, 2006 and hereby enclosed in the Evidence Appendix) and J. Buchmann (see page 131 of Buchmann J. Introduction to Cryptography, Springer-Verlag, New York, Berlin, 2004, previously submitted in the response filed on August 22, 2006 and hereby enclosed in the Evidence Appendix).

In addition, Schneier does not explain nor describe that the conversion operation *f* is a permutation operation. In Fig. 12.1, Schneier presents a general scheme of conversions of algorithm DES as a sequence of performing sixteen typical conversion rounds and designates one conversion round as the conversion operation *f*. Then, in Fig. 12.2, Schneier discloses elementary operations constituting each round. Schneier detailed elementary operations constituting the conversion operation *f* in the sections immediately following page 270 and Fig. 12.1, such as the section of "The Expansion Permutation" on page 273. Because this section immediately follows the description of the general scheme of the make-up of algorithm, Schneier clearly does not indicate that this section describes consecutive elementary operations, which specify the conversion operation *f*. Apparently, this circumstance misled the Examiner. The fact that the conversion operation *f* in algorithm DES is a composite operation and includes a sequence of elementary conversion operations, follows from the description of DES provide also in the widely known document -- Menezes (see Menezes A.J., Vanstone S.A., Handbook of Applied Cryptography, CRC Press, 1996), on page 255 (Figs. 7.10)(copy enclosed in Evidence Appendix). The document Menezes clearly discloses that peforming the conversion operation *f* is a sequence of the following elementary operations: i) subblock R_{i-1} expansion operation (e.g. of subblock R_0), ii) modulo 2 bit-by-bit summation operation performed on an expanded data subblock and subkey K_i (e.g. subkey K_I), iii) substitution operation performed on the result obtained at the output of the preceding operation, and iv) fixed bit permutation operation. Here, the fixed bit permutation operation does not depend on any data subblock being converted. The same fixed bit permutation is performed for all possible input data in a given round when performing the operation *f* at step iv). The document J. Buchmann also indicates that the conversion step operation *f* used in algorithm DES consists of the above elementary operations i), ii), iii) and iv), on page 131 (Buchmann J. Introduction to Cryptography, Springer-Verlag, New York, Berlin, 2004, copy enclosed in Evidence Appendix). Therefore, the conversion operation *f* is not a bit permutation operation and does not include any bit permutation operation dependent on a data subblock. Accordingly, algorithm DES does not anticipate the claimed invention.

Furthermore, although page 270 of Schneier indicates that "The right half of the data is expanded to 48 bits via an expansion permutation, combined with 48 bits of a shifted and permuted key via an Xor ...", one of ordinary skill in the art cannot conclude from this reference that a bit permutation operation was previously performed on the subkey **depending on** some **data subblock** being converted. On page 272 and in section "The Key Transformation", Schneier discloses, what bit permutation operation was performed on the subkey: "First, the 56-bit key is divided into two 28-bit halves. Then, the halves are circularly shifted left by either or two bits, depending on the round." Thus, in algorithm DES, the **bit permutation operation** is performed on the key by **depending on the number of the round, but not on the data subblock, i.e. algorithm DES lacks the feature of performing the subkey bit permutation operation depending on the data subblock being converted**, that is presented in the claimed invention.

The Examiner has established a connection between a fixed permutation performed on a key to form the predetermined encryption round and the data block which is converted at the predetermined encryption round. Applicant respectfully disagrees with such interpretation of DES encryption algorithm that is well known to a person of ordinary skill in the art, since Schneier does not describe such an operation, because it is not used in DES algorithm. A connection arbitrarily established by the Examiner between the round key, e.g., for the i-th round, and the subblock value at the i-th round does not relate to the technical content of the conversion procedures and operations used in DES algorithm as understood by a person of ordinary skill in the art.

The fact that, in DES algorithm, round keys are formed regardless of data converted, is supported by the following obvious facts:

1. Independently of data converted, at each predetermined encryption round, the same round key is used.
2. All round keys can be calculated in advance, i.e. before the point when a data block for encryption will be selected for encryption with the DES algorithm.

These facts are supported by well-known literature references describing DES algorithm, such as those enclosed in Evidence Appendix. These references describe the procedure of forming DES round keys as an independent one and not depending on data converted. A person of ordinary skill in the art who is familiar with block encryption methods can confirm this position. Thus, in algorithm DES, the bit permutation operation is performed on the key depending on the number of the round and not on the data subblock, i.e. the feature of performing the subkey bit permutation operation depending on the data subblock being converted, that is present in the claimed invention, is novel.

Therefore, the currently presented claims are not anticipated by Schneier and the rejection under 35 U.S.C. § 102 (b) has been overcome. Accordingly, withdrawal of the rejection under 35 U.S.C. § 102 (b) is respectfully requested.

Respectfully submitted,

JACOBSON HOLMAN PLLC

By____________________
John C. Holman
Registration No. 22,769

Date: May 8, 2007
(202) 638-6666
400 Seventh Street, N.W.
Washington, D.C. 20004

Enclosed:
CLAIM APPENDEX
EVIDENCE APPENDIX
RELATED PROCEEDING APPENDEX

VIII. CLAIM APPENDEX

Claim 1 (previously presented): A method for block encryption of discrete data, comprising the steps of: generating an encryption key in the form of a set of subkeys, breaking down a data block into $N \geq 2$ data subblocks and alternately converting said data subblocks by performing a two-place operation on the data subblock and the subkey, wherein, prior to carrying out said two-place operation on an i-th data subblock and a subkey, an operation of permuting subkey bits is performed on the subkey depending on the value of a j-th data subblock, where $i \neq j$.

Claim 2 (cancelled)

Claim 3 (previously presented): The method according to claim 1, wherein an operation of cyclic offsetting subkey bits depending on the value of the j-th data subblock is used as the j-th data subblock-dependent operation of permuting subkey bits.

Claim 4 (cancelled)

Claim 5 (previously presented) The method according to claim 1, wherein the operation of permuting subkey bits is performed on one of said set of subkeys depending on the value of the j-th data subblock, where $i \neq j$, and the value of another subkey.

IX. EVIDENCE APPENDIX

1. Bruce Schneier, Applied Cryptography, 1996, John Wiley & Sons, pages 270 – 273, cited by Examiner in Office Actions mailed on October 19, 2005 and September 14, 2006;

2. Menezes A.J., Vanstone S.A., Handbook of Applied Cryptography, CRC Press, 1996, page 225, submitted by Applicant in Response filed on August 22, 2006 in response to Office Action of May 23, 2006;

3. Buchmann J. Introduction to Cryptography, Springer-Verlag, New York, Berlin, 2004, page 131, submitted by Applicant in Response filed on August 22, 2006 in response to Office Action of May 23, 2006;

Evidence Appendix 1

APPLIED CRYPTOGRAPHY, SECOND EDITION

PROTOCOLS, ALGORITHMS, AND SOURCE CODE IN C

BRUCE SCHNEIER

John Wiley & Sons, Inc.
New York • Chichester • Brisbane • Toronto • Singapore

Publisher: Katherine Schowalter
Editor: Phil Sutherland
Assistant Editor: Allison Roarty
Managing Editor: Robert Aronds
Text Design & Composition: North Market Street Graphics

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc. is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

This text is printed on acid-free paper.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional person should be sought.

In no event will the publisher or author be liable for any consequential, incidental, or indirect damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising from the use or inability to use the protocols and algorithms in this book, even if the publisher or author has been advised of the possibility of such damages.

Some of the protocols and algorithms in this book are protected by patents and copyrights. It is the responsibility of the reader to obtain all necessary patent and copyright licenses before implementing in software any protocol or algorithm in this book. This book does not contain an exhaustive list of all applicable patents and copyrights.

Some of the protocols and algorithms in this book are regulated under the United States Department of State International Traffic in Arms Regulations. It is the responsibility of the reader to obtain all necessary export licenses before implementing in software for export any protocol or algorithm in this book.

Library of Congress Cataloging-in-Publication Data:

Schneier, Bruce
Applied Cryptography Second Edition : protocols, algorithms, and source code in C
/ Bruce Schneier.
p. cm.
Includes bibliographical references (p. 675).
ISBN 0-471-12845-7 (cloth : acid-free paper). — ISBN
0-471-11709-9 (paper : acid-free paper)
1. Computer security. 2. Telecommunication—Security measures.
3. Cryptography. I. Title.
QA76.9.A25S35 1996
005.8'2—dc20 95-12398
CIP

Printed in the United States of America
10

[1191], the algorithm was recertified for another five years [1150]. Software implementations of DES were finally allowed to be certified.

Anyone want to guess what will happen in 1998?

12.2 DESCRIPTION OF DES

DES is a block cipher; it encrypts data in 64-bit blocks. A 64-bit block of plaintext goes in one end of the algorithm and a 64-bit block of ciphertext comes out the other end. DES is a symmetric algorithm: The same algorithm and key are used for both encryption and decryption (except for minor differences in the key schedule).

The key length is 56 bits. (The key is usually expressed as a 64-bit number, but every eighth bit is used for parity checking and is ignored. These parity bits are the least-significant bits of the key bytes.) The key can be any 56-bit number and can be changed at any time. A handful of numbers are considered weak keys, but they can easily be avoided. All security rests within the key.

At its simplest level, the algorithm is nothing more than a combination of the two basic techniques of encryption: confusion and diffusion. The fundamental building block of DES is a single combination of these techniques (a substitution followed by a permutation) on the text, based on the key. This is known as a **round**. DES has 16 rounds; it applies the same combination of techniques on the plaintext block 16 times (see Figure 12.1).

The algorithm uses only standard arithmetic and logical operations on numbers of 64 bits at most, so it was easily implemented in late 1970s hardware technology. The repetitive nature of the algorithm makes it ideal for use on a special-purpose chip. Initial software implementations were clumsy, but current implementations are better.

Outline of the Algorithm

DES operates on a 64-bit block of plaintext. After an initial permutation, the block is broken into a right half and a left half, each 32 bits long. Then there are 16 rounds of identical operations, called Function f, in which the data are combined with the key. After the sixteenth round, the right and left halves are joined, and a final permutation (the inverse of the initial permutation) finishes off the algorithm.

In each round (see Figure 12.2), the key bits are shifted, and then 48 bits are selected from the 56 bits of the key. The right half of the data is expanded to 48 bits via an expansion permutation, combined with 48 bits of a shifted and permuted key via an XOR, sent through 8 S-boxes producing 32 new bits, and permuted again. These four operations make up Function f. The output of Function f is then combined with the left half via another XOR. The result of these operations becomes the new right half; the old right half becomes the new left half. These operations are repeated 16 times, making 16 rounds of DES.

If B_i is the result of the ith iteration, L_i and R_i are the left and right halves of B_i, K_i is the 48-bit key for round i, and f is the function that does all the substituting and permuting and XORing with the key, then a round looks like:

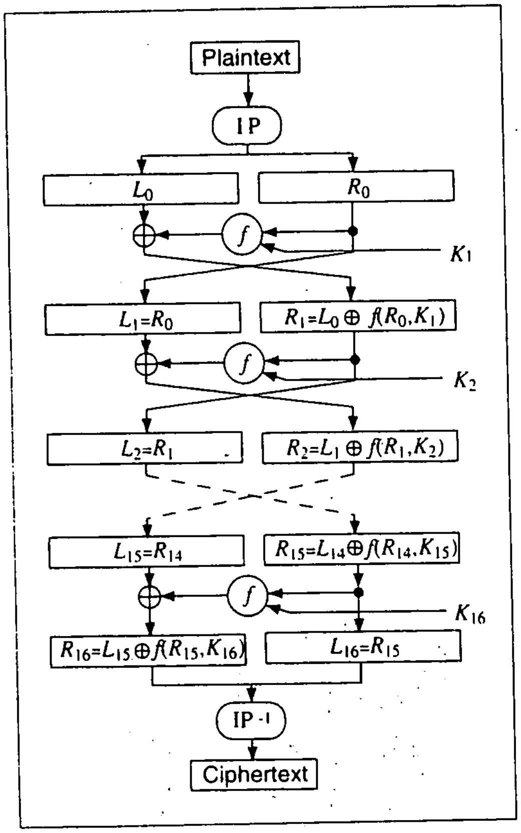

Plaintext
IP
L_0
R_0
f
K_1
$L_1=R_0$
$R_1=L_0 \oplus f(R_0,K_1)$
f
K_2
$L_2=R_1$
$R_2=L_1 \oplus f(R_1,K_2)$
$L_{15}=R_{14}$
$R_{15}=L_{14} \oplus f(R_{14},K_{15})$
f
K_{16}
$R_{16}=L_{15} \oplus f(R_{15},K_{16})$
$L_{16}=R_{15}$
IP^{-1}
Ciphertext

Figure 12.1 DES.

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

The Initial Permutation

The initial permutation occurs before round 1; it transposes the input block as described in Table 12.1. This table, like all the other tables in this chapter, should be read left to right, top to bottom. For example, the initial permutation moves bit 58 of the plaintext to bit position 1, bit 50 to bit position 2, bit 42 to bit position 3, and so forth.

The initial permutation and the corresponding final permutation do not affect DES's security. (As near as anyone can tell, its primary purpose is to make it easier to load plaintext and ciphertext data into a DES chip in byte-sized pieces. Remember that DES predates 16-bit or 32-bit microprocessor busses.) Since this bit-wise permutation is difficult in software (although it is trivial in hardware), many software implementations of DES leave out both the initial and final permutations. While this new algorithm is no less secure than DES, it does not follow the DES standard and should not be called DES.

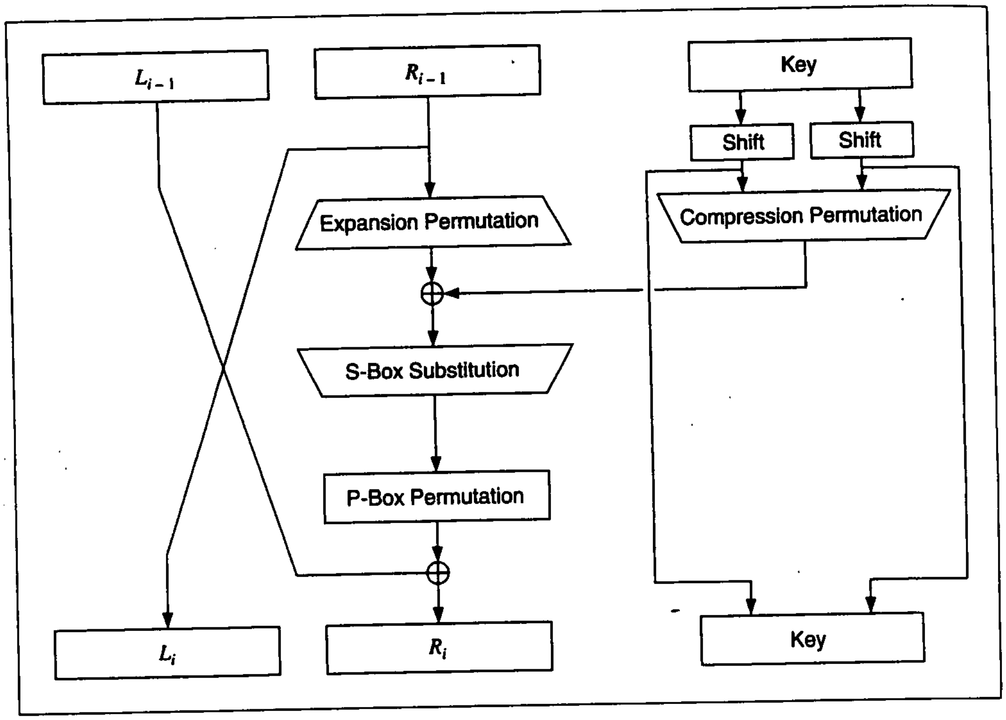


Figure 12.2 One round of DES.

The Key Transformation

Initially, the 64-bit DES key is reduced to a 56-bit key by ignoring every eighth bit. This is described in Table 12.2. These bits can be used as parity check to ensure the key is error-free. After the 56-bit key is extracted, a different 48-bit **subkey** is generated for each of the 16 rounds of DES. These subkeys, K_i, are determined in the following manner.

First, the 56-bit key is divided into two 28-bit halves. Then, the halves are circularly shifted left by either one or two bits, depending on the round. This shift is given in Table 12.3.

Table 12.1
Initial Permutation

58,	50,	42,	34,	26,	18,	10,	2,	60,	52,	44,	36,	28,	20,	12,	4,
62,	54,	46,	38,	30,	22,	14,	6,	64,	56,	48,	40,	32,	24,	16,	8,
57,	49,	41,	33,	25,	17,	9,	1,	59,	51,	43,	35,	27,	19,	11,	3,
61,	53,	45,	37,	29,	21,	13,	5,	63,	55,	47,	39,	31,	23,	15,	7

Table 12.2
Key Permutation

57,	49,	41,	33,	25,	17,	9,	1,	58,	50,	42,	34,	26,	18,
10,	2,	59,	51,	43,	35,	27,	19,	11,	3,	60,	52,	44,	36,
63,	55,	47,	39,	31,	23,	15,	7,	62,	54,	46,	38,	30,	22,
14,	6,	61,	53,	45,	37,	29,	21,	13,	5,	28,	20,	12,	4

After being shifted, 48 out of the 56 bits are selected. Because this operation permutes the order of the bits as well as selects a subset of bits, it is called a **compression permutation**. This operation provides a subset of 48 bits. Table 12.4 defines the compression permutation (also called the permuted choice). For example, the bit in position 33 of the shifted key moves to position 35 of the output, and the bit in position 18 of the shifted key is ignored.

Because of the shifting, a different subset of key bits is used in each subkey. Each bit is used in approximately 14 of the 16 subkeys, although not all bits are used exactly the same number of times.

The Expansion Permutation

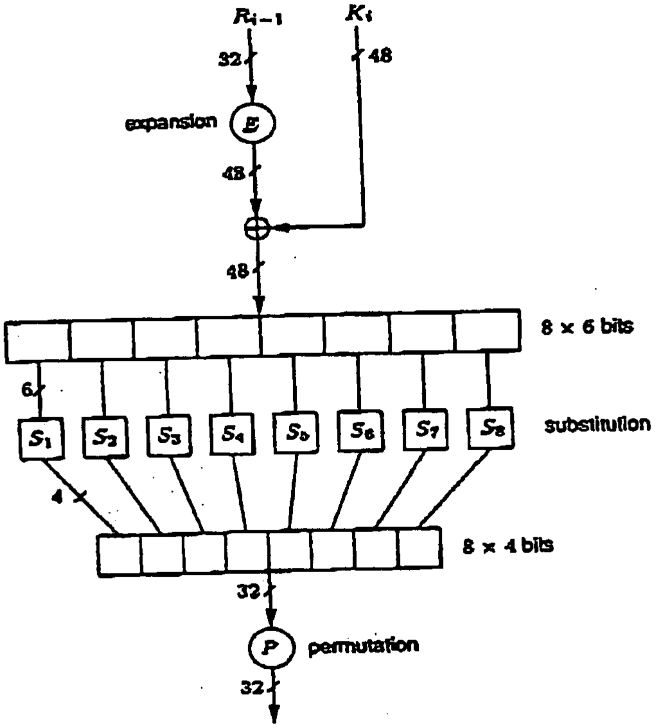
This operation expands the right half of the data, R_i, from 32 bits to 48 bits. Because this operation changes the order of the bits as well as repeating certain bits, it is known as an **expansion permutation**. This operation has two purposes: It makes the right half the same size as the key for the XOR operation and it provides a longer result that can be compressed during the substitution operation. However, neither of those is its main cryptographic purpose. By allowing one bit to affect two substitutions, the dependency of the output bits on the input bits spreads faster. This is called an **avalanche effect**. DES is designed to reach the condition of having every bit of the ciphertext depend on every bit of the plaintext and every bit of the key as quickly as possible.

Figure 12.3 defines the expansion permutation. This is sometimes called the **E-box**. For each 4-bit input block, the first and fourth bits each represent two bits of the output block, while the second and third bits each represent one bit of the output block. Table 12.5 shows which output positions correspond to which input positions. For example, the bit in position 3 of the input block moves to position 4 of the output block, and the bit in position 21 of the input block moves to positions 30 and 32 of the output block.

Although the output block is larger than the input block, each input block generates a unique output block.

Table 12.3
Number of Key Bits Shifted per Round

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1



$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

Figure 7.10: DES inner function f.

7.83 Algorithm DES key schedule

INPUT: 64-bit key $K = k_1 \dots k_{64}$ (including 8 odd-parity bits).
OUTPUT: sixteen 48-bit keys K_i, $1 \le i \le 16$.

1. Define v_i, $1 \le i \le 16$ as follows: $v_i = 1$ for $i \in \{1, 2, 9, 16\}$; $v_i = 2$ otherwise. (These are left-shift values for 28-bit circular rotations below.)
2. $T \leftarrow$ PC1(K); represent T as 28-bit halves (C_0, D_0). (Use PC1 in Table 7.4 to select bits from K: $C_0 = k_{57}k_{49} \dots k_{36}$, $D_0 = k_{63}k_{55} \dots k_4$.)
3. For i from 1 to 16, compute K_i as follows: $C_i \leftarrow (C_{i-1} \hookleftarrow v_i)$, $D_i \leftarrow (D_{i-1} \hookleftarrow v_i)$, $K_i \leftarrow$ PC2(C_i, D_i). (Use PC2 in Table 7.4 to select 48 bits from the concatenation $b_1b_2 \dots b_{56}$ of C_i and D_i: $K_i = b_{14}b_{17} \dots b_{32}$. '$\hookleftarrow$' denotes left circular shift.)

If decryption is designed as a simple variation of the encryption function, savings result in hardware or software code size. DES achieves this as outlined in Note 7.84.

7.84 Note (*DES decryption*) DES decryption consists of the encryption algorithm with the same key but reversed key schedule, using in order $K_{16}, K_{15}, \dots, K_1$ (see Note 7.85). This works as follows (refer to Figure 7.9). The effect of IP^{-1} is cancelled by IP in decryption, leaving (R_{16}, L_{16}); consider applying round 1 to this input. The operation on the left half yields, rather than $L_0 \oplus f(R_0, K_1)$, now $R_{16} \oplus f(L_{16}, K_{16})$ which, since $L_{16} = R_{15}$ and $R_{16} = L_{15} \oplus f(R_{15}, K_{16})$, is equal to $L_{15} \oplus f(R_{15}, K_{16}) \oplus f(R_{15}, K_{16}) = L_{15}$. Thus round 1 decryption yields (R_{15}, L_{15}), i.e., inverting round 16. Note that the cancellation

Evidence Appendix 3

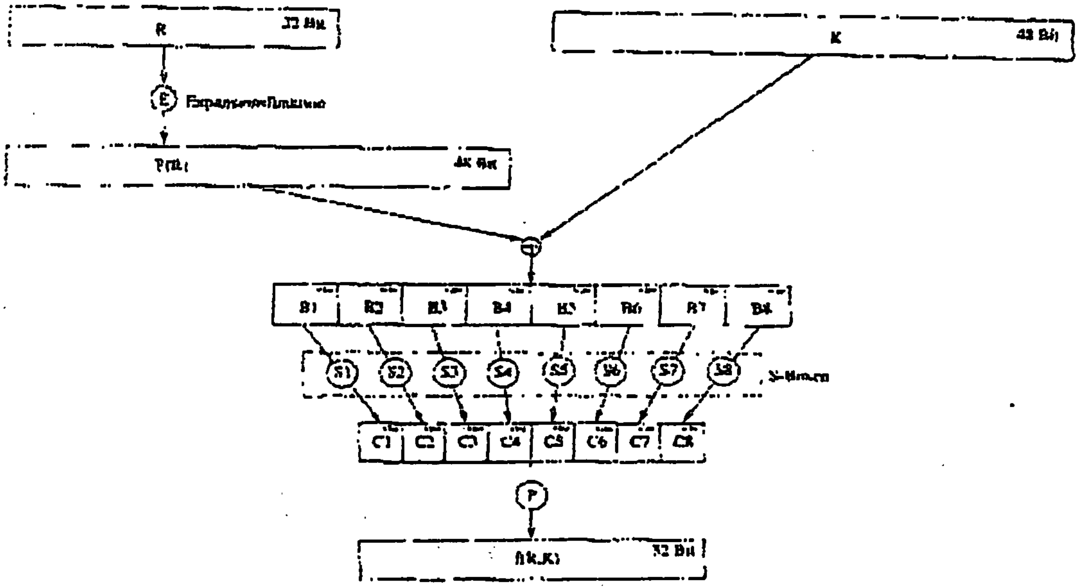


FIGURE 5.1 The f-function of DES.

is computed with $B_i \in \{0,1\}^6$, $1 \leq i \leq 8$. In the next step, functions

$$S_i : \{0,1\}^6 \rightarrow \{0,1\}^4, \quad 1 \leq i \leq 8$$

are used (the so-called S-boxes). They are described below. Using those functions, the string

$$C = C_1C_2C_3C_4C_5C_6C_7C_8$$

is determined, where $C_i = S_i(B_i)$, $1 \leq i \leq 8$. It has length 32. The permutation P from Table 5.3 is applied to this string. The result is $f_K(R)$.

5.2.4 S-boxes

Now we describe the S-boxes S_i, $1 \leq i \leq 8$. They are the heart of DES because they are highly nonlinear (see Exercise 5.5.6). They are shown in Table 5.4.Each S-box is represented by a table with four rows and 16 columns. For each string $B = b_1b_2b_3b_4b_5b_6$, the value $S_i(B)$ is computed as follows. The integer with binary expansion b_1b_6 is used as the row index. The integer with binary expansion $b_2b_3b_4b_5$ is used as the column index. The entry of the S-box in this row and column is written in binary expansion. This expansion is padded with leading zeros such that its length is four. The result is $S_i(B)$.

X. RELATED PROCEEDING APPENDEX

There are no related proceedings.